ALIGNFLOW: MULTI-AXIS CACHE SCHEDULING FOR EFFICIENT CONTINUOUS VIDEO EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion transformers now dominate video generation, and training-free feature caching is the standard way to cut their denoising cost, yet every existing cache is designed and evaluated for a single generation call. Real editing is not a single call: users issue a sequence of cumulative edits, each conditioned on the previous output. This discards the redundancy that exists across consecutive edits and makes approximation error compound recursively, so single-turn fidelity cannot characterize a session. Extending acceleration across edits is not a matter of adding a second cache. Raw cross-edit reuse is intermittent and, used alone, needs far more exact evaluations than a step-axis cache; finer token-level reuse traded a 4.22% latency reduction for a 52.55% memory increase in our diagnostic; and the edited region is precisely where past information is invalid. We present ALIGNFLOW, a training-free controller that predicts the classifier-free-guided output and schedules reuse jointly along the step, region, and edit axes. Damped first-order extrapolation of the guided residual supplies the acceleration backbone; a progress-aligned bank of the previous turn's exact outputs supplies a conservatively mixed cross-edit source, which a dilated latent mask forbids inside the edited region; and a closed-loop calibrator admits each source only when its measured per-region, per-phase reliability clears a threshold, otherwise falling back to an exact step. On five-turn editing chains with VACE Wan2.1-1.3B, ALIGNFLOW reaches a $1.965\times$ speedup at 27.230 dB PSNR, 8.972 dB above the strongest accelerated baseline at lower latency, and transfers to real video and a second architecture. Session-average gains exceed final-turn gains, so we report per-turn fidelity and drift together rather than claiming that recursive drift is removed.

1 INTRODUCTION

Video diffusion transformers produce high-quality video at a high price: every output requires tens of full denoiser evaluations (Peebles & Xie, 2023; Wan et al., 2025; Kong et al., 2024). Training-free feature caching has become the default remedy, reusing or forecasting intermediate results across denoising steps (Liu et al., 2025a; Zhou et al., 2025; Liu et al., 2025b). These methods share an unstated assumption: the unit of work is one generation call. Editing breaks that assumption. A user who makes a ball blue, adds a stripe, adds a star, makes the star glossy and then adds dust runs five calls in sequence, and each call starts from the previous output. Two consequences follow. There is redundancy *across* calls that a single-call cache cannot see, and any approximation error made at one turn becomes the input of the next, so a cache that looks harmless on one call can drift over a session.

Exploiting this setting is less direct than it appears (Section 3.1): cross-edit reuse alone is intermittent and needs about $1.8\times$ the exact evaluations of a step-axis cache, token-level reuse saves little latency at a large memory cost, and every method drifts over a session. The step axis must therefore supply acceleration, the region axis must decide where each source may be trusted, and the edit axis must be an auxiliary source that earns admission.

ALIGNFLOW (Figure 1) is a training-free controller built on these observations. It predicts the classifier-free-guided (CFG) output (Ho & Salimans, 2022) rather than internal features, with a damped extrapolation of the guided residual as its backbone; it mixes in a small, calibrated contribution from the previous turn's exact outputs, but only outside the edited region; and it admits each

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

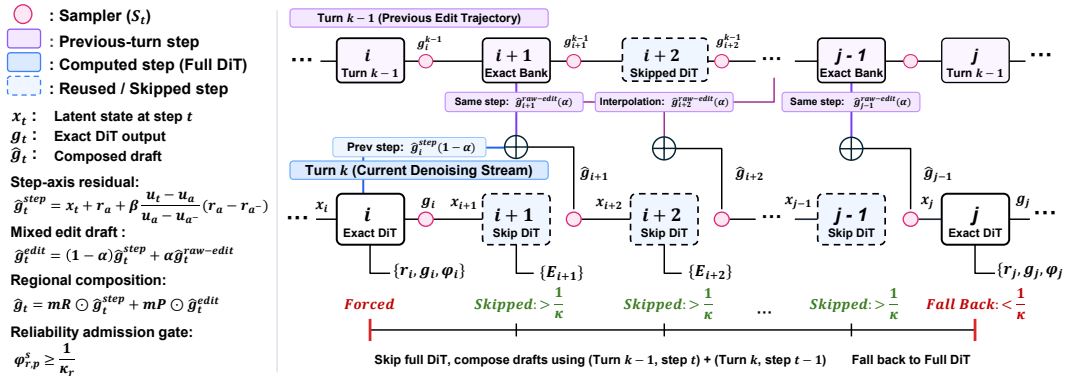


Figure 1: **Overview of ALIGNFLOW.** In turn k (bottom stream), exact DiT steps record the residual r, the guided output g and reliability scores φ. Skipped steps compose a draft from the within-turn step predictor $\widehat{g}^{\text{step}}$ in the edited region R and from a mixture with the previous turn's exact bank, $\widehat{g}^{\text{edit}}$, in the preserved region P. A step is skipped only when every nonempty region has a source whose reliability score clears that region's threshold $1/\kappa_r$; otherwise the full DiT is evaluated. Forced anchors (warmup, phase boundaries, periodic and final steps) refresh the predictors.

prediction source only when its measured reliability, tracked separately per region and per denoising phase, clears a threshold, falling back to an exact step otherwise.

Contributions. (i) We formulate continuous video editing as an acceleration problem with three reuse axes (step, region, edit) and show empirically why each axis needs a different role. (ii) We propose ALIGNFLOW, which combines a damped guided-residual predictor, a region-constrained cross-edit source with conservative mixing, and a per-region, per-phase calibrator with exact fallback, without modifying model weights. (iii) On Wan2.1 ALIGNFLOW reaches $1.965\times$ with 27.230 dB PSNR, ahead of TeaCache, HetCache and EasyCache in both latency and fidelity; it is the fastest method on two real-video datasets and on MiniMax-H3. (iv) We report where the method is weaker: on real video some baselines retain higher fidelity, session-average gains exceed final-turn gains, and a staleness guard mitigates but does not remove ten-turn drift.

2 RELATED WORK

Video diffusion and editing. Diffusion models (Ho et al., 2020; Rombach et al., 2022) moved from U-Net video generators (Ho et al., 2022; Blattmann et al., 2023b;a) to diffusion transformers (Peebles & Xie, 2023) such as Sora, Open-Sora, CogVideoX, HunyuanVideo and Wan2.1 (Brooks et al., 2024; Zheng et al., 2024; Yang et al., 2025; Kong et al., 2024; Wan et al., 2025), often trained with rectified flow (Liu et al., 2022). Diffusion video editing ranges from tuning-free video-to-video pipelines (Ku et al., 2024) and instruction-driven editing (Cheng et al., 2024) to unified frameworks such as VACE (Jiang et al., 2025), which encode text, image, video and mask inputs through one conditioning interface and make mask-localized, instruction-conditioned editing practical. Real usage is multi-turn, yet acceleration research treats each edit as an independent generation.

Training-free acceleration and feature caching. Faster samplers (Song et al., 2020; Lu et al., 2022; Karras et al., 2022) reduce the step count, while token merging (Bolya & Hoffman, 2023), sparse attention (Xi et al., 2025) and attention-guided token pruning (Wang et al., 2024) reduce the cost per step. Feature caching exploits redundancy between steps, either with fixed or precomputed schedules (Ma et al., 2024; Zhao et al., 2025; Selvaraju et al., 2024; Chen et al., 2024; Wimbauer et al., 2024; Lyu et al., 2025) or with runtime criteria (Liu et al., 2025a; Zhou et al., 2025; Kahatapitiya et al., 2025; Ye et al., 2024). A spatial line caches or refreshes computation per token or region (Zou et al., 2024; Zou et al.; Liu et al., 2026b), and HetCache (Liu et al., 2026a) applies token-level context caching to masked video editing. Forecasting methods predict features by Taylor expansion (Liu et al., 2025b), refine the forecast with a predictor-corrector scheme (Zheng et al., 2026), or verify it speculatively (Liu et al., 2025c). All of these operate within one generation. ALIGNFLOW adds

the edit axis, couples it with the edit's spatial locality, and treats cross-edit reuse as an auxiliary, calibrated source rather than a new backbone.

3 PRELIMINARIES

A session consists of K cumulative edits. Given the previous output $v^{(k-1)}$, an instruction c_k and an optional spatial mask M_k, turn k produces $v^{(k)}$ through T denoising steps. With latent $x_t^{(k)}$ and denoiser output $y_{b,t}^{(k)}$ for CFG branch $b \in \{u, c\}$, the sampler uses

$$g_t^{(k)} = y_{u,t}^{(k)} + \gamma\big(y_{c,t}^{(k)} - y_{u,t}^{(k)}\big), \qquad x_{t+1}^{(k)} = \mathcal{S}_t\big(x_t^{(k)}, g_t^{(k)}\big), \tag{1}$$

where $\mathcal{S}_t$ is the sampler update including any solver state. An *exact step* evaluates the denoiser at the current latent; a *cached step* substitutes a predicted guided output. An exact step on an accelerated trajectory is not the same as the corresponding step of an independent unaccelerated session, and because each turn conditions on the preceding output, the two sessions diverge recursively. All fidelity numbers below compare against such an independent unaccelerated session, so they include this propagated divergence.

3.1 EMPIRICAL OBSERVATIONS THAT SHAPE THE DESIGN

Three measurements, all on VACE Wan2.1-1.3B, determine what each reuse axis should do.

Step prediction must be the backbone. Used alone, the previous turn's trajectory rarely substitutes for a denoising step: in the isolated-axis diagnostic of Section 5.5 it is reused on only 12 of 600 decisions at T1, 236 at T2 and 165 at T5. Over the diagnostic, a cross-edit-only controller therefore needs $2{,}223$ exact evaluations where a step-axis cache needs $1{,}253$. Cross-edit information is useful as a second source on top of a step predictor, not as a replacement for it.

Finer granularity costs memory. In a two-chain diagnostic (Table 9 in Section A.1), adding token-level key/value reuse cut editing latency by 4.22% but raised peak CUDA allocation by 52.55%, or 8.11 GiB, with no measurable quality change. We therefore route at the region level and obtain compute savings by skipping whole steps. The two-case sample does not establish a universal tradeoff; it fixes an implementation choice.

Single-turn fidelity does not describe a session. Every method we evaluate loses fidelity over five turns, including ALIGNFLOW (Figure 2), and a method's T1-to-T5 drop depends on where it starts: TeaCache-fast falls from 18.885 to 15.149 dB, ALIGNFLOW from 31.848 to 23.152 dB, a larger drop from a far higher level. Drop and final-turn fidelity must be read together, and we report both throughout.

4 METHOD

ALIGNFLOW predicts CFG-guided outputs and validates the predictions at exact anchors. A cache hit skips a complete denoising step; if any nonempty region requires exact evaluation, the full DiT is executed and its output is used everywhere. No pretrained weight is changed.

4.1 DAMPED PREDICTION ALONG THE DENOISING TRAJECTORY

Predictions are indexed by normalized scheduler progress $u_t \in [0, 1]$. At an exact anchor a we record the residual $r_a = g_a - x_a$ in guided-output coordinates. With the two latest anchors $a^- < a$, we extrapolate to first order and shrink the increment toward the most recent residual:

$$\widehat{g}_t^{\,\mathrm{step}} = x_t + r_a + \beta\,\frac{u_t - u_a}{u_a - u_{a^-}}\,(r_a - r_{a^-}), \qquad \beta = 0.25. \tag{2}$$

With a single anchor the residual is held constant. The subtraction defines a prediction coordinate and does not assume that the denoiser is an identity-plus-residual network; all admission decisions operate on the reconstructed guided output. Damping limits the extrapolated increment but does not guarantee a lower error on every step. Warmup, phase-boundary, periodic and final steps are exact and refresh the predictor (schedule in Section A.4); reliability is tracked separately in three progress phases.

4.2 CROSS-EDIT FORECASTING WITH CONSERVATIVE MIXING

A trajectory bank retains the guided outputs evaluated exactly during the preceding turn, and linear interpolation in progress gives a reference $\bar{g}_{k-1}(u)$ between stored anchors. Early exact probes of the current turn measure the drift and a similarity:

$$d_i = g_{a_i}^{(k)} - \bar{g}_{k-1}(u_{a_i}), \qquad \sigma_k = \cos\big(g_{a_0}^{(k)}, \bar{g}_{k-1}(u_{a_0})\big). \tag{3}$$

When $\sigma_k > \sigma_{\min}$ and enough probes exist, a first-order drift polynomial $D_k(u)$ fitted to the latest observations yields the raw cross-edit source, which is mixed conservatively with the step source:

$$\begin{aligned} \widehat{g}_t^{\,\mathrm{raw}} &= \bar{g}_{k-1}(u_t) + \eta_k D_k(u_t), \qquad \eta_k = \mathrm{clip}\Big(\tfrac{\sigma_k - \sigma_{\min}}{1-\sigma_{\min}}, 0, 1\Big), \\ \widehat{g}_t^{\,\mathrm{edit}} &= (1-\alpha)\,\widehat{g}_t^{\,\mathrm{step}} + \alpha\,\widehat{g}_t^{\,\mathrm{raw}}, \qquad \alpha = 0.05. \end{aligned} \tag{4}$$

The bank stores the preceding *accelerated* turn's exact evaluations, not a separate unaccelerated trajectory, and progress normalization aligns lookups without guaranteeing transfer across sampling protocols. The bank replaces the previous turn's entries after each completed turn; with N_a anchors of D values it occupies about $2N_aD$ bytes in FP16, so trajectory storage is $O(TD)$ and independent of session length.

4.3 REGIONAL SOURCE CONSTRAINTS AND EXACT FALLBACK

An external mask, resized to the latent grid and dilated, defines the edited region R_k and the preserved region $P_k = \Omega \setminus R_k$; without one, an early-probe drift-energy quantile provides a routing mask. The admissible sources are

$$\mathcal{A}(R_k) = \{\mathrm{step}\}, \qquad \mathcal{A}(P_k) = \{\mathrm{step}, \mathrm{edit}\}, \tag{5}$$

with uncalibrated candidates removed, so cross-edit information never enters the edited region. When both nonempty regions admit a candidate, their outputs are composed by the masks, $\widehat{g}_t = m_R \odot \widehat{g}_t^{\,\mathrm{step}} + m_P \odot \widehat{g}_t^{\,\mathrm{edit}}$, and the DiT step is skipped; otherwise the exact output is used. Routing therefore selects sources and decides skips; it does not implement token-selective attention. Preserved-region activity at exact observations can tighten that region's threshold and expand the edited boundary (Section A.5). The controller branches on whether an external mask was supplied, never on dataset identity.

4.4 CLOSED-LOOP CALIBRATION AND SOURCE ADMISSION

Before an eligible exact evaluation the controller forms every available draft; after obtaining g_t it measures their regional relative squared error and updates an error score q and a reliability score φ for each source s, region r and phase p:

$$\begin{aligned} E_{r,t}^{s} &= \frac{\|m_r \odot (\widehat{g}_t^{\,s} - g_t)\|_2^2}{\max(\|m_r \odot g_t\|_2^2, \epsilon)}, \\ q_{r,p}^{s} \leftarrow \lambda q_{r,p}^{s} + (1-\lambda)\,\mathrm{clip}\big(E_{r,t}^{s}/B_t, q_{\min}, q_{\max}\big), &\qquad \varphi_{r,p}^{s} = \mathrm{clip}\big(1/q_{r,p}^{s}, \varphi_{\min}, \varphi_{\max}\big). \end{aligned} \tag{6}$$

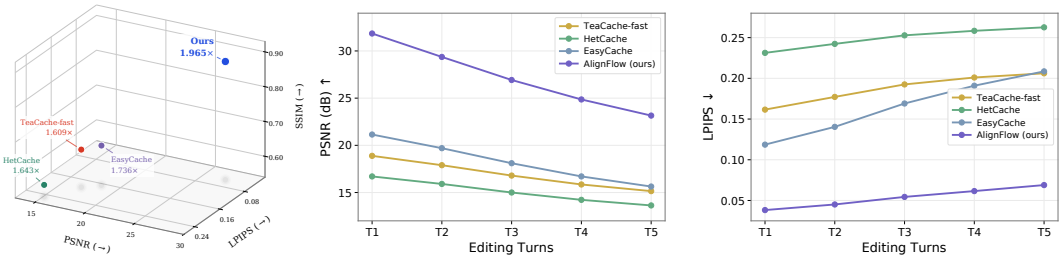
A source is admitted when $\varphi_{r,p}^{s} \geq 1/\kappa_r$, i.e. when its normalized error score is at most the regional prior κ_r. Among admitted candidates the one with the lowest error score $q_{r,p}^{s}$ wins, except that the preserved region prefers the mixed edit source when its error score is within $1.25\times$ of the best; if nothing is admissible, an exact step is requested. The priors are empirical admission levels chosen on pilot runs; the edited-region priors are more permissive because that region's content is being regenerated and its fidelity is dominated by the edit itself, whereas the preserved region is where drift is most visible. Budget anchors and reliability-triggered fallbacks update calibration; warmup, probe and reset evaluations do not. Step statistics persist across turns; edit statistics reset with each new edit. The calibrator estimates reliability from observed anchors: it does not measure the error of a skipped step and does not bound final-video error, which is why we evaluate drift explicitly (Section A.7).

5 EXPERIMENTS

Setup. Four experimental blocks are used. **Wan100**: 100 VBench-derived (Huang et al., 2024) chains from eight content categories, each with one initial generation and five cumulative edits. **Real video**:

Table 1: **Continuous editing on VACE Wan2.1-1.3B**, averaged over all chains and five editing turns (Wan100). Best and second among accelerated methods.

	Efficiency		Visual retention			
Method	Latency (s) ↓	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓	Prompts
Exact / Official	74.281	1.000×	∞	1.000	0.000	100
TeaCache-fast	46.179	1.609×	16.916	0.648	0.188	100
HetCache	45.211	1.643×	15.095	0.574	0.249	100
EasyCache	42.799	1.736×	18.258	0.654	0.166	100
ALIGNFLOW (ours)	**37.798**	**1.965×**	**27.230**	**0.880**	**0.054**	100



(a) Speed and fidelity.

(b) Per-turn PSNR and LPIPS.

Figure 2: **Wan100 results.** (a) Each method in PSNR, LPIPS and SSIM, labelled with its speedup over Exact. (b) Fidelity to the unaccelerated session at each editing turn. Every method drifts as accelerated outputs are fed back; ALIGNFLOW starts higher and stays higher at every turn.

20 LOVEU TGVE videos (Wu et al., 2023) with full-frame edits and 20 DAVIS videos (Perazzi et al., 2016) with foreground masks, each extended into a constructed five-turn session. **Architecture transfer**: 20 chains on MiniMax-H3 (FL2VA, 33B, W4A8). **Controlled studies**: 20 VBench chains on Wan2.1 for the axis ablation (five turns) and the staleness study (ten turns). Wan runs use VACE Wan2.1-1.3B (Jiang et al., 2025; Wan et al., 2025) at 832×480, 49 frames, 30 steps, CFG scale 5.0; H3 uses 640×352, 124 frames, 20 steps. In every protocol turn $k{+}1$ consumes the same method's turn-k output. Baselines are TeaCache-fast (Liu et al., 2025a), HetCache (Liu et al., 2026a) and EasyCache (Zhou et al., 2025), with matched prompts, seeds, masks and sampling. Latency is synchronized CUDA generation time excluding preprocessing and evaluation; speedup is the ratio of mean latencies against the matched Exact run. Fidelity (PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018)) is measured against the unaccelerated session at the same prompt, seed and turn, on 16 sampled frames (Wan100), 24 (H3) or all 49 (real video and controlled studies). Unless marked, results are fixed-seed means; details are in Sections A.8 and A.9.

5.1 LARGE-SCALE COMPARISON ON WAN2.1

ALIGNFLOW is both the fastest and the most faithful accelerated method on Wan100 (Table 1): it cuts latency from 74.281 to 37.798 seconds per edit ($1.965\times$) and exceeds the strongest baseline, EasyCache, by 8.972 dB PSNR and 0.226 SSIM while reducing LPIPS by 0.112. Figure 2 shows that this gap is not a first-turn effect. At T1 ALIGNFLOW is about 10.7 dB above EasyCache and at T5 still about 7.5 dB above, and its LPIPS at T5 (0.069) is lower than any baseline's LPIPS at T1. The baselines are single-call caches with step-skip thresholds tuned for one generation; applied to a chain, their errors are compounded by conditioning. This is consistent with all three starting below 22 dB even at the first edit, since T1 already conditions on each method's own accelerated initial generation.

Two trends are visible per turn (Table 2). Speed peaks at T2 ($2.280\times$) and then declines to $1.767\times$ at T5, which is consistent with the calibrator admitting fewer skips as the inputs of later turns diverge further from the exact session. Fidelity decreases monotonically, from 31.848 to 23.152 dB. A larger absolute PSNR drop than a baseline's is not a worse result when every turn stays higher:

Table 2: **ALIGNFLOW by editing turn on Wan100.** Turn 0 (initial generation) is excluded.

Turn	Latency (s) ↓	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓
T1	37.898	1.962×	**31.848**	**0.903**	**0.038**
T2	**32.630**	**2.280**×	29.365	0.893	0.045
T3	36.767	2.022×	26.923	0.881	0.054
T4	39.735	1.867×	24.860	0.868	0.062
T5	41.957	1.767×	23.152	0.854	0.069

Table 3: **Five-turn continuous editing of real video.** Session latency is the sum over T1 to T5; fidelity averages all edited outputs of each dataset. The drop column is descriptive: a method with poor T1 fidelity can show a small drop. Best and second among accelerated methods per dataset.

	Efficiency		Visual retention			Session drift	
Method	Session (s) ↓	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR@5 ↑	T1→T5 drop
LOVEU TGVE (20 videos, full-frame edits, no mask)							
Exact	210.993	1.000×	∞	1.000	0.000	∞	n/a
EasyCache	157.265	1.342×	21.432	0.812	0.100	16.244	14.006
TeaCache-fast	147.859	1.427×	22.346	0.844	0.088	17.423	13.671
HetCache	158.203	1.334×	**23.534**	**0.874**	**0.068**	**18.691**	13.054
ALIGNFLOW (ours)	**146.809**	**1.437**×	22.356	0.850	0.081	17.623	12.888
DAVIS (20 videos, foreground masks)							
Exact	420.164	1.000×	∞	1.000	0.000	∞	n/a
EasyCache	246.997	1.701×	25.472	0.902	0.045	21.751	9.233
TeaCache-fast	273.810	1.535×	**32.932**	**0.957**	**0.013**	**29.083**	8.564
HetCache	254.473	1.651×	27.482	0.927	0.035	23.201	10.103
ALIGNFLOW (ours)	**235.515**	**1.784**×	32.280	0.954	0.015	28.520	8.507

TeaCache-fast falls by 3.7 dB from a starting point 13 dB lower. We therefore read drop and final-turn fidelity together.

5.2 CONTINUOUS EDITING OF REAL VIDEOS

Real video is where ALIGNFLOW's advantage narrows, and the reason is informative (Table 3). It remains the fastest method on both datasets (1.437× on LOVEU, 1.784× on DAVIS), but it is not the most faithful on either. On LOVEU, HetCache is 1.2 dB closer to Exact at 1.334×, and ALIGNFLOW's 1.05 s lead over TeaCache-fast is 0.71% of session time, not a statistically established margin. On DAVIS, TeaCache-fast is 0.65 dB closer to Exact but 38.3 s slower per session, whereas ALIGNFLOW is 11.5 s faster than the next fastest baseline and 6.8 dB more faithful than it. Controller traces show how differently the two datasets use the axes. On LOVEU, which has no external mask, the edit source is selected 1,164 times in total while whole-step skipping stays at 43.3% to 47.7% of steps. On DAVIS, which provides masks, the edit source is selected 294 times at T1 and 19 at T2, then never again, while whole-step skipping continues at 56% to 59% of steps. The controller thus keeps its step-axis acceleration after cross-edit reuse stops being reliable, which is the intended fallback, but it also means that on masked real footage the cross-edit axis contributes mainly in the first two turns.

5.3 TRANSFER TO A SECOND ARCHITECTURE

On MiniMax-H3, which differs in latent representation, frame count and schedule, ALIGNFLOW cuts latency from 72.798 to 44.590 seconds (1.633×; Table 4). It is 16.2% faster than HetCache with better fidelity on all three metrics, and 35.5% faster than TeaCache-fast, which is 0.49 dB closer to Exact but barely accelerates (1.053×). ALIGNFLOW adds only 0.043 GiB of peak memory, and its no-reference VBench-8 score (Huang et al., 2024) stays within 0.0004 of Exact. Unlike on Wan100, its speedup grows across turns, from 1.371× at T1 to 1.811× at T5, while PSNR falls from 37.230 to 27.919 dB; per-turn efficiency therefore does not track per-turn fidelity in a fixed way across

Table 4: **Continuous editing on MiniMax-H3** (20 chains, five turns). The EasyCache row comes from a separate matched run without memory or VBench measurements. VBench-8 is a no-reference quality score.

Method	Latency (s) ↓	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓	Peak (GiB) ↓	VBench-8 ↑
Exact / Official	72.798	1.000×	∞	1.000	0.000	5.254	0.745539
TeaCache-fast	69.106	1.053×	**32.593**	**0.949**	**0.015**	5.815	0.745017
HetCache	53.224	1.368×	31.277	0.944	0.020	6.002	0.744433
EasyCache	49.429	1.473×	25.316	0.838	0.069	n/a	n/a
ALIGNFLOW (ours)	**44.590**	**1.633**×	32.101	0.948	0.018	**5.297**	**0.745180**

Table 5: **Incremental axis decomposition** on VACE Wan2.1-1.3B (20 chains, five turns, all 49 frames). Cache is controller storage beyond the unaccelerated run. Fixed-seed means without paired intervals.

Configuration	Latency (s) ↓	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR@5 ↑	Peak (GiB)	Cache (GiB)
Step-axis backbone only	41.688	1.788×	25.554	0.847	0.131	21.278	**14.570**	0.092
+ mask-aware region routing	38.445	1.939×	25.574	0.849	0.131	21.281	**14.570**	**0.068**
+ cross-edit propagation	**38.330**	**1.945**×	27.606	0.873	0.059	21.372	14.644	0.184
+ cross-axis calibration (full)	38.338	**1.945**×	**28.613**	**0.891**	**0.047**	**22.405**	14.644	0.184

architectures. Audio, which H3 also generates, is less well preserved (correlation 0.900 with Exact audio, SI-SDR 12.18 dB), and HetCache and TeaCache-fast retain higher audio similarity; transfer of video-side acceleration does not imply multimodal dominance.

5.4 AXIS-WISE DECOMPOSITION

The three axes play different roles (Table 5). The step-axis backbone supplies the main acceleration ($1.788\times$). Region routing cuts a further 7.78% of latency with unchanged fidelity. By design it replaces one global admission test with per-region tests under region-specific priors, so a step can be skipped when each region's source is reliable enough for that region even where a single global threshold would have forced an exact step. Cross-edit propagation barely changes latency (-0.30%) but raises average PSNR by 2.032 dB and cuts LPIPS by 55%, and calibration adds 1.007 dB and cuts LPIPS by another 20% at a recorded cost of 0.008 s. The fidelity gain from cross-edit propagation is large for a source mixed at only 5%, and this ablation does not separate how much comes from the mixed draft itself and how much from the changed admission decisions that its calibrated scores induce; we report the net effect. Two limits apply. The final-turn gain is much smaller than the average gain (PSNR@5 $+1.127$ dB versus $+3.059$ dB on average), so the axes improve the session far more than they cure its end state. And the numbers are fixed-seed operating points without paired intervals; a separate isolated-axis diagnostic with bootstrap intervals is reported in Section 5.5.

5.5 ISOLATING CROSS-EDIT PROPAGATION

The incremental ablation measures cross-edit propagation on top of the step backbone; the isolated diagnostic asks what the edit axis does alone (Table 6). Runtime counters confirm the configuration: the cross-edit-only run records no step decisions, no region events and 777 edit decisions. It is closer to Exact in pooled PSNR, SSIM, LPIPS and MSE, but 46.5% slower. It is slower on all 20 prompts. In a paired prompt-level bootstrap with 50,000 resamples (Table 10), its MSE reduction of $+0.001565$ has a 95% interval of $[0.000105, 0.003233]$, which excludes zero, whereas its pooled-PSNR improvement of $+0.827$ dB has an interval of $[-0.485, 2.142]$, which does not. The control disables step prediction, region routing and cross-axis calibration while keeping cross-edit propagation; its raw forecast remains subject to exact-anchor and admission logic, so disabling calibration does not remove every reliability check. The study passed an independent audit of all prompt-turn outputs and evaluated frame pairs. The comparison is between two compute budgets, $2{,}223$ versus $1{,}253$ exact decisions, so its better fidelity is at least partly explained by more exact computation rather than by a more accurate predictor. Thirteen T1 outputs are pixel-identical to Exact, which is why pooled PSNR is reported. Even with this extra computation, pooled PSNR falls from

Table 6: **Isolated step axis versus isolated cross-edit axis** (20 chains, five turns, 49 frames). $\text{PSNR}_{\text{pool}}$ is $-10\log_{10}$ of the mean MSE and stays finite with pixel-identical videos. Each row makes 3,000 decisions.

Method	Latency (s)↓	Speedup↑	$\text{PSNR}_{\text{pool}}$↑	SSIM↑	LPIPS↓	MSE↓	Exact matches	Exact	Step	Edit
Step-axis only	**41.688**	**1.788**×	24.009	0.847	0.131	0.003973	0	**1253**	1747	0
Cross-edit only	61.078	1.221×	**26.185**	**0.856**	**0.091**	**0.002407**	13	2223	0	777

Table 7: **Ten-turn study** on VACE Wan2.1-1.3B (20 chains). The two fixed-propagation rows differ only in the staleness guard, which forces a full exact turn after five consecutive propagated turns. Turn-10 MSE, SSIM, LPIPS and memory are in Table 11 (Section A.3).

Configuration	Latency (s)↓	Speedup↑	Resets	PSNR↑	SSIM↑	LPIPS↓	PSNR@5↑	PSNR@10↑
Exact, $K = 10$	83.776	1.000×	0	∞	1.000	0.000	∞	∞
Full, $K = 10$	45.138	1.856×	0	23.705	0.867	0.059	22.405	17.212
Fixed propagation, guard	47.677	1.757×	12	**23.764**	**0.869**	**0.058**	22.405	**17.338**
Fixed propagation, no guard	**44.781**	**1.871**×	0	23.721	0.868	0.059	22.405	17.265

40.950 to 22.426 dB across the five turns: reusing the previous turn does not by itself stop session drift. Evidence for the *incremental* benefit of the edit axis comes from Table 5, where it improves fidelity at nearly unchanged latency.

5.6 Session length and the staleness guard

Over ten turns the full controller keeps a $1.856\times$ speedup while fidelity falls from 35.768 dB at T1 to 17.212 dB at T10 (Table 7): per-turn acceleration is stable, and recursive drift is not. The default controller stops selecting the edit source after its early turns in this workload, so its staleness cap never activates; to measure the guard we use a fixed-propagation control and toggle only the guard. It performs 12 exact reset turns and improves T10 PSNR by 0.073 dB, MSE by 0.000196 and SSIM by 0.002 at a cost of 2.896 s per turn ($1.871\times \rightarrow 1.757\times$), and it doubles the extra cache from 0.184 to 0.363 GiB (Table 11). Peak memory is identical across the three accelerated configurations. The guard mitigates staleness but cannot erase inherited error: a fully exact turn still conditions on the accelerated previous video, which is why a reset refreshes the cache without resetting the session (Section A.7). Formally, if the unaccelerated turn map is L_k-Lipschitz in its video input and the accelerated turn adds at most δ_k error at the same input, the session error obeys $e_k \le L_k e_{k-1} + \delta_k$, so a cache-free turn sets $\delta_k = 0$ but leaves $L_k e_{k-1}$. All outputs of this study decoded to 49 frames and passed prompt, seed, edit-type and source-chain checks.

5.7 Where the exact budget goes

The decision counters give a direct view of what the controller spends. The step-axis backbone evaluates 1,253 of 3,000 steps exactly (41.8%, or 12.5 per 30-step turn), most of them scheduled anchors rather than fallbacks (Section A.4). This floor, not the predictor, bounds the attainable speedup of the backbone: with 12.5 of 30 steps exact its denoiser-only ideal is $2.39\times$, against a measured $1.788\times$, the gap being overhead outside the denoiser. Region routing lifts the measured speedup to $1.945\times$ by admitting skips that a single global test would reject. The edit source changes *which* draft fills a skipped step more than *whether* a step is skipped. In the single-session pilot (Table 8), the step-only and edit-enabled runs make 112 and 107 exact steps; the edit source is selected 76 times but removes only five of them, and in the incremental ablation it improves PSNR by 2.032 dB while changing latency by 0.30%. On real video, whole-step skipping stays at 43% to 59% of steps, and on DAVIS it continues at that level after the edit source stops being admitted. The practical consequence is that the three axes are not additive in speed: the step and region axes set the latency, and the edit axis and calibration set the fidelity at that latency.

5.8 Case study

Figure 3 shows a representative five-turn session side by side, illustrating where the per-turn fidelity differences of Figure 2 become visible. Over the session AlignFlow reaches $1.59\times$ against $1.29\times$

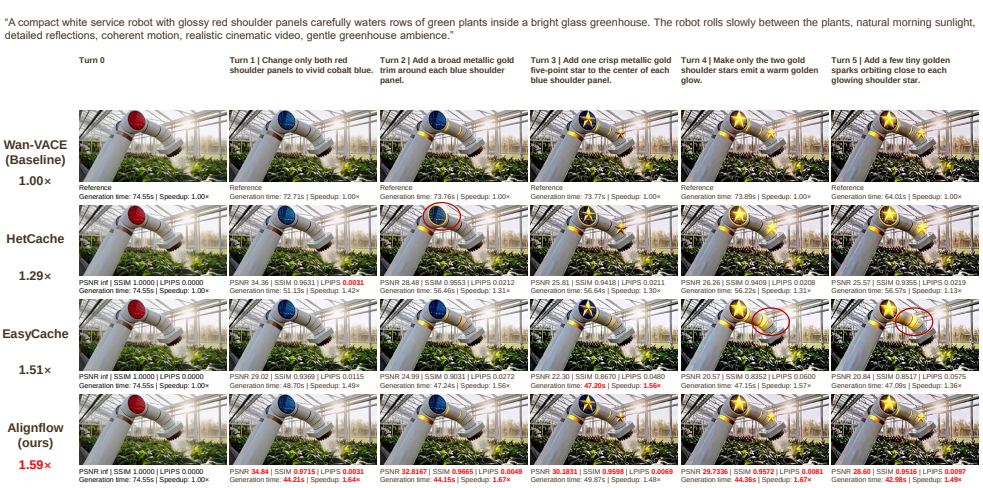


Figure 3: **Qualitative continuous-editing session.** Greenhouse robot: T0 and edits T1–T5 for Exact (Wan-VACE), HetCache, EasyCache and ALIGNFLOW, with per-turn fidelity, generation time and speedup in each panel. Red circles mark visible deviations from Exact.

for HetCache and $1.51\times$ for EasyCache. At T5 it retains 28.60 dB PSNR and 0.0097 LPIPS, against 25.57 dB and 0.0219 for HetCache and 20.84 dB and 0.0575 for EasyCache. The circled deviations of the baselines accumulate on the edited shoulder panels, the region that each turn modifies and later turns build on. Two further sessions are in Section A.11.

Evaluation scope. PSNR, SSIM and LPIPS measure fidelity to the matched unaccelerated session, not instruction following or preference. Most results use one fixed seed, so small differences without intervals are descriptive. The VBench content categories are prompt subjects, not the five native VACE task types, so the evaluation is not a per-task analysis of inpainting, outpainting or reference-guided editing. Quality still degrades over long sessions, especially for global reconstruction and large-area changes, and the staleness guard only mitigates this.

6 CONCLUSION AND OUTLOOK

Continuous video editing should not be treated as a sequence of unrelated generation calls. In ALIGNFLOW, a step-axis predictor removes most of the computation within each turn, region routing removes more by testing reliability where it matters, and the previous turn's exact outputs, used conservatively outside the edited region and admitted only when calibrated, raise session fidelity at almost no latency cost. On Wan2.1 it reaches $1.965\times$ with 8.972 dB higher PSNR than the strongest accelerated baseline, and it is the fastest method on two real-video datasets and on the 33B MiniMax-H3 model, where it adds only 0.043 GiB of peak memory; on real video, some baselines nonetheless retain higher fidelity. The advantage holds turn by turn: on Wan2.1 the per-turn speedup stays between $1.767\times$ and $2.280\times$, and the fifth-turn LPIPS of ALIGNFLOW remains below every baseline's first-turn LPIPS. The ablation makes the division of labour explicit: the step and region axes set the latency, while the edit axis and calibration set the fidelity at that latency.

The results also mark what remains open. Session-average gains exceed final-turn gains, and over ten turns fidelity still falls from 35.768 to 17.212 dB while per-turn acceleration stays stable, so recursive drift is mitigated rather than removed. The most direct next step is to spend exact computation according to accumulated session error, through adaptive resets, uncertainty-aware source selection and semantic change detection that trigger exact turns only when drift justifies them. On masked real footage the edit source is admitted mainly in the first two turns, so keeping cross-edit reuse reliable deeper into a session is a second target. A third is to extend the bank beyond the immediately preceding turn, so that edits which revisit earlier states can reuse them, and a fourth is to pair region-level routing with memory-efficient token-level reuse once the memory overhead observed in our diagnostic can be controlled. Since most of our results are fixed-seed means, multi-seed evaluation with paired intervals would sharpen these comparisons.

AI USE STATEMENT

Generative AI tools were used solely for manuscript preparation, specifically language polishing and organization of text written by the authors. All AI-assisted text was reviewed and approved by the authors.

ETHICS STATEMENT

ALIGNFLOW is a training-free inference controller that changes no pretrained weight. The experiments use VBench-derived prompts, LOVEU TGVE and DAVIS videos, and the VACE Wan2.1-1.3B and MiniMax-H3 models. All reported fidelity is measured by automatic metrics against the matched unaccelerated session; these metrics measure retention rather than instruction following or preference (Section 5.8).

REPRODUCIBILITY STATEMENT

We submit the code as part of the supplementary materials. The controller is specified in Section 4, with the parameters used in all Wan experiments in Table 12, forecasting, storage and calibration details in Section A.5, and pseudocode in Section A.6. Workloads, seed bases, hardware and software versions for the real-video and H3 runs, and all baseline settings are given in Section A.8; fidelity and drift conventions in Section A.9; and the prompt format with example chains in Section A.10.

REFERENCES

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22563–22575. IEEE, 2023b.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *2023 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 4599–4603. IEEE, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. Δ-DiT: A training-free acceleration method tailored for diffusion transformers. *arXiv preprint arXiv:2406.01125*, 2024.

Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. In *International Conference on Learning Representations*, volume 2024, pp. 16867–16879, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202. IEEE, 2025.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15240–15252. IEEE, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363. IEEE, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15853–15863. IEEE, 2025b.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Fei Ren, Shaobo Wang, Kaixin Li, and Linfeng Zhang. Speca: Accelerating diffusion transformers with speculative feature caching. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10024–10033, 2025c.

Tianyi Liu, Ye Lu, Linfeng Zhang, Chen Cai, Jianjun Gao, Yi Wang, Kim-Hui Yap, and Lap-Pui Chau. Accelerating diffusion-based video editing via heterogeneous caching: Beyond full computing at sampled denoising timestep. *arXiv preprint arXiv:2603.24260*, 2026a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2346–2356, 2026b.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Zhengyao Lyu, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K Wong. Fastercache: Training-free video diffusion model acceleration with high quality. In *International Conference on Learning Representations*, volume 2025, pp. 33132–33156, 2025.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772. IEEE, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16080–16089. IEEE, 2024.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6211–6220. IEEE, 2024.

Jay Zhangjie Wu, Xiuyu Li, Difei Gao, Zhen Dong, Jinbin Bai, Aishani Singh, Xiaoyu Xiang, Youzeng Li, Zuwei Huang, Yuanxi Sun, et al. Cvpr 2023 text guided video editing competition. *arXiv preprint arXiv:2310.16003*, 2023.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Hancheng Ye, Jiakang Yuan, Renqiu Xia, Xiangchao Yan, Tao Chen, Junchi Yan, Botian Shi, and Bo Zhang. Training-free adaptive diffusion with bounded difference approximation strategy. *Advances in Neural Information Processing Systems*, 37:306–332, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *International Conference on Learning Representations*, volume 2025, pp. 3296–3319, 2025.

Shikang Zheng, Liang Feng, Xinyu Wang, Qinming Zhou, Peiliang Cai, Chang Zou, Jiacheng Liu, Yuqi Lin, Junjie Chen, Yue Ma, et al. Forecast then calibrate: Feature caching as ode for efficient diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13449–13457, 2026.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Xin Zhou, Dingkang Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860*, 2025.

Chang Zou, Evelyn Zhang, Runlin Guo, Haohang Xu, Conghui He, Xuming Hu, and Linfeng Zhang. Accelerating diffusion transformers with dual feature caching, 2024. *URL https://arxiv.org/abs/2412.18911*.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

A Appendix

A.1 Pilot comparisons

The matched step/edit pilot uses VACE Wan2.1-1.3B, 832×480, 49 frames, 30 steps, one fixed five-turn sequence and base seed 12345; both configurations disable the CFG-branch cache.

Table 8: **Single-session pilot.** Wall time includes overhead outside denoising and is not comparable with the generation latency of the main tables.

Configuration	Exact steps	Skipped steps	CFG forward calls	Wall time (s)	PSNR ↑	SSIM ↑	LPIPS ↓
Step-only	112	38	224	482.219	31.389	0.943	0.024
Edit-enabled	**107**	**43**	**214**	**475.423**	**31.879**	**0.947**	**0.021**

The edit-enabled run records 76 edit-source selections but only five fewer exact steps: a regional source selection need not save a model invocation. The pilot supports the feasibility of an additional source; one sequence cannot establish the marginal gain of the 5% mixture. Table 9 reports the token K/V diagnostic of Section 3.1. Neither pilot measures adjacent-step correlation directly, and we make no claim about a universal correlation law.

Table 9: **Token K/V reuse diagnostic** (two chains, `animal_pomeranian_soccer` and `animal_curious_cat`, five edits each). Means over the ten edits.

Configuration	Editing latency (s) ↓	Peak CUDA (GiB) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Without token K/V	40.801	**15.441**	23.795	0.835	0.105
With token K/V	**39.080**	23.555	23.808	0.835	0.104

A.2 Isolated-axis diagnostic

Table 10 gives the paired bootstrap summarized in Section 5.5.

A.3 Ten-turn study checks

The two fixed-propagation rows of Table 7 coincide at T5 because the guard can only activate after five consecutive propagated turns.

Table 10: **Paired prompt-level bootstrap** for Table 6 (50,000 resamples). Positive values favor cross-edit only. The pooled-PSNR difference is the mean of per-prompt pooled-PSNR differences, which is not the difference of the global pooled values in Table 6 because PSNR is nonlinear in MSE.

Metric	Mean difference	95% confidence interval
Latency improvement (s)	-19.390	$[-22.412, -18.429]$
Pooled-PSNR improvement (dB)	$+0.827$	$[-0.485, 2.142]$
MSE reduction	$+0.001565$	$[0.000105, 0.003233]$

Table 11: **Turn-10 fidelity and memory** for the configurations of Table 7. Cache is controller storage beyond the unaccelerated run.

Configuration	PSNR ↑	MSE ↓	SSIM ↑	LPIPS ↓	Peak (GiB)	Cache (GiB)
Exact, $K = 10$	∞	0.000000	1.000	0.000	14.570	0.000
Full, $K = 10$	17.212	0.024261	0.779	0.107	14.644	0.184
Fixed propagation, guard	**17.338**	**0.023923**	**0.782**	**0.106**	14.644	0.363
Fixed propagation, no guard	17.265	0.024119	0.780	0.107	14.644	0.184

Table 12: **Controller parameters** used in all Wan experiments.

Parameter	Value or rule
Step and edit polynomial order	1 and 1
Guided-residual damping β	0.25
Edit mixture coefficient α	0.05 (fixed by equal bounds)
Probe indices	0, 1, 2 (zero-based)
Warmup exact steps	first 4
Similarity threshold $\sigma_{\min}$	0.70
Maximum consecutive propagated turns	5
Strict full-turn staleness reset	enabled
Terminal exact steps	last 3
Maximum distance from an exact anchor	8 steps
Periodic schedule with external mask	every 5 steps, offset 2
Periodic schedule without external mask	every 4 steps, offset 0
Phase boundaries	progress $1/3$ and $2/3$, first step of each phase exact
Base reference error budget	0.02
Budget control points	$(1.0, 1.0, 1.001)$, interpolated over progress
EMA decay λ	0.85
Ratio and inverse-ratio clipping	$[0.05, 20]$
Initial step and edit ratios	1.0 and 1.0
Minimum edit observations per phase	1
Preserved-region prior κ_P	1.35, reduced by activity monitoring
Edited-region prior κ_R	2.5 with external mask, 4.5 without
Near-best edit preference factor	1.25
Mask dilation	radius 1 on the latent grid
Inferred-mask drift-energy quantile	0.85
Preserved-token activity tolerance	relative squared energy > 0.05
Activity expansion trigger	0.95
Preserved-prior contraction coefficient c	0.50
Guided bank storage	FP16 on GPU, every exact step
Token K/V and CFG-branch cache	disabled

A.4 CONTROLLER PARAMETERS

Scheduler progress is $u_t = (\tau_t - \tau_0)/(\tau_{T-1} - \tau_0)$. Periodic anchors that would fall too close to another inserted anchor can be suppressed. Under the 30-step protocol the two nominal schedules contain 12 and 14 exact steps before reliability-triggered fallbacks, a protocol-specific count.

A.5 FORECASTING, STORAGE AND CALIBRATION DETAILS

For order one, the step predictor uses Lagrange extrapolation through the two latest distinct progress points and the edit predictor an equivalent first-order Newton divided difference; degenerate anchor sets fall back to a lower order. The implementation contains a clipped least-squares routine for the edit mixture, but with equal bounds α is fixed and only the mixture's error score is adapted. The edit candidate is not trusted until calibrated for the current turn and phase. For preserved-region activity $a \in [0, 1]$ the prior is adjusted to

$$\kappa_P^{\mathrm{cur}} = 1 + (\kappa_P - 1)\big[1 - c \min(1, a/a_{\mathrm{trig}})\big], \qquad c = 0.5, \tag{7}$$

computed from the current exact output and the progress-matched bank. If activity exceeds the trigger, active tokens adjacent to the edited boundary can be absorbed into the edited mask; this is not an unrestricted expansion. The admission test $\varphi \geq 1/\kappa_r$ is applied once, and B_t is a normalization scale, not a per-step error bound.

A.6 CONTROLLER PSEUDOCODE

```
At the start of a turn:
    reset within-turn forecasts and current edit-source calibration
    keep learned step-source calibration and the preceding trajectory
    ↪ bank
    initialize regions; select the schedule by external-mask availability
    force a full exact turn if the staleness counter requires it

For each denoising step:
    build the available step and cross-edit drafts
    if a forced anchor or reset applies:
        request a full exact step
    else:
        admit legal sources using their phase and region scores
        edited region: step source; preserved region: step or 5% edit
        ↪ mixture
        request a full exact step if any nonempty region has no source
    if an exact step is requested:
        run the original CFG denoiser at the current latent
        update probes, forecasts, and scores (budget anchors and
        ↪ fallbacks only)
        return the exact guided output
    else:
        compose regional drafts and skip the denoiser
    apply the original sampler update

At the end of a turn:
    replace the bank with this turn's exact evaluations
    increment the propagation counter if an edit source was used, else
    ↪ reset it
```

A.7 WHY A SESSION-DRIFT GUARANTEE IS NOT CLAIMED

Anchor calibration observes draft error at the current accelerated latent; it does not verify skipped outputs against their counterfactual exact values or compare latents with the independent unaccelerated session. Let F_k be the unaccelerated turn map, $\widetilde{F}_k$ its accelerated version and $e_k = \|\widetilde{v}_k - v_k\|$. If F_k is L_k-Lipschitz in its video input and the accelerated turn adds at most δ_k error at the same input,

then adding and subtracting $F_k(\widetilde{v}_{k-1})$ and recursing gives

$$e_k \le L_k e_{k-1} + \delta_k, \qquad e_K \le \Big(\prod_{j=1}^{K} L_j\Big) e_0 + \sum_{i=1}^{K} \delta_i \prod_{j=i+1}^{K} L_j. \tag{8}$$

A cache-free turn sets $\delta_k = 0$ but leaves $L_k e_{k-1}$: a reset does not erase inherited error unless the map contracts uniformly or truly returns to the reference state, which are additional assumptions. The regional errors of Equation (6) do not supply the δ_k. In the five-edit benchmark a guard reset could only occur after the final turn, so that benchmark cannot test guarantees based on resets.

A.8 Workloads, seeds and timing

Wan100 contains 20 existing chains and 80 additional ones, 10 per category (animal, architecture, food, human, lifestyle, plant, scenery, vehicle); the initial stages are excluded from the edit averages, and references are matched by case and stage. LOVEU cases have no external mask (full-frame conditioning, inferred routing mask); DAVIS cases provide foreground masks. Candidate and baseline sessions evolve independently. The negative-prompt text cache of AlignFlow is disabled because the baselines recompute the encoding each turn. For N cases and K edits,

$$L^m_{\text{edit}} = \frac{1}{NK} \sum_{i,k} \ell^m_{i,k}, \qquad L^m_{\text{session}} = \frac{1}{N} \sum_{i,k} \ell^m_{i,k}, \qquad S^m = L^{\text{exact}}/L^m, \tag{9}$$

a ratio of means; per-turn speedups divide that turn's exact mean by the method's. Real-video and H3 runs use an NVIDIA RTX PRO 6000 Blackwell Server Edition (PyTorch 2.8.0 with CUDA 12.8 and PyTorch 2.11.0 with CUDA 13.0 respectively), deterministic algorithms, TF32 disabled, model offloading and the T5 text encoder (Raffel et al., 2020) on CPU; seed bases are 20260901 (real video), 20260830 (H3) and 20260810 (Wan100), plus the case index. Baseline settings are EasyCache threshold 0.05, TeaCache-fast threshold 0.10 and HetCache delta 0.08 on Wan; on H3, TeaCache-fast uses threshold 0.10 with five retention steps and an H3-specific calibration polynomial, and HetCache uses delta 0.02, full threshold 0.03, 16 context clusters, context retention 0.7 and EMA factor 0.5.

A.9 Fidelity and drift conventions

Evaluators decode videos to RGB, pair uniformly spaced frames and resize to the reference size if needed; PSNR and SSIM (Wang et al., 2004) use the 0 to 255 range, and LPIPS (Zhang et al., 2018) uses AlexNet at 256×256 in $[-1, 1]$. The exact self-reference is reported as PSNR ∞, SSIM 1, LPIPS 0, and its drop is undefined (n/a); averages over accelerated videos use finite PSNR values. The isolated-axis diagnostic reports pooled PSNR because 13 of its videos are pixel-identical to Exact. The drop is $\overline{\text{PSNR}}_{T1} - \overline{\text{PSNR}}_{T5}$; because methods start at different T1 fidelity, a small drop can reward a method that is already inaccurate, so final-turn PSNR and LPIPS are always reported beside it. On Wan100, the no-reference VBench-8 score (Huang et al., 2024) of AlignFlow is 0.709 over all six stages and 0.705 over the five edits.

A.10 Five-turn prompt format and examples

Each session has one base prompt and five ordered edits; each turn receives the preceding output and one text condition describing the desired cumulative state. For Wan100 a case stores `base_prompt` and five `edits` (`edit_type`, `instruction`, `target_prompt`); the runner passes `target_prompt` directly. Edit types vary by case (object and attribute additions, appearance, color, lighting, effect, weather and motion changes); there is no universal template. H3 chains share the Wan100 base descriptions and edits and wrap them in a six-field template:

```
subject_definitions: <Video 1> is the current source video for the same
↪  continuous editing session. <Audio 1> is its synchronized soundtrack.
summary: [video editing + audio reuse] {instruction}
retention_analysis: <Video 1> is partially_preserved. Preserve all
↪  subjects, identities, object geometry, actions, motion trajectories,
↪  timing, camera movement, framing, background, and lighting except the
↪  property explicitly changed by the instruction. <Audio 1> is
↪  fully_copy.
```

```
detailed_description: Reproduce the source shot coherently and apply only
↪  this requested edit: {target} Do not reset or remove modifications
↪  made in earlier turns.
overall_soundscape: Fully copy <Audio 1>.
non_diegetic_music: Fully copy <Audio 1>.
```

Table 13: **Example chain shared by Wan100 and H3.**

Turn	Incremental instruction	Cumulative target
T0	Initial generation	a cute pomeranian dog playing with a soccer ball
T1	Make the soccer ball bright blue.	A cute Pomeranian dog is playing with a bright blue soccer ball; keep the dog, setting, motion, and camera unchanged.
T2	Add one thin white stripe to the blue soccer ball.	A cute Pomeranian dog is playing with the same bright blue soccer ball with one thin white stripe; keep the dog, setting, motion, and camera unchanged.
T3	Add a small gold star to the striped ball.	A cute Pomeranian dog is playing with the same bright blue striped soccer ball with one small gold star; keep everything else unchanged.
T4	Make the gold star slightly glossy.	A cute Pomeranian dog is playing with the same bright blue striped soccer ball whose small gold star is slightly glossy; keep everything else unchanged.
T5	Add a few subtle dust particles near the rolling ball.	A cute Pomeranian dog is playing with the same decorated soccer ball with a few subtle dust particles nearby; keep the dog, setting, motion, and camera unchanged.

For LOVEU, T1 applies the official object-edit target and T2 to T5 successively add a thin white detail, a subtly glossy finish, a soft warm highlight and faint dust motes, retaining earlier changes. For DAVIS, T1 combines the scene description with a case-specific foreground change and T2 to T5 apply the same four refinements. Both append the preservation suffix [P]: *Preserve all other subjects, actions, motion trajectories, timing, camera movement, framing, background, and lighting.* These five-turn protocols are constructed for evaluation; they are not native multi-turn annotations of LOVEU, DAVIS or VBench.

Table 14: **Real-video examples**: LOVEU `cat-in-the-sun` and DAVIS `bike-packing` ([P] abbreviated).

Turn	LOVEU cumulative prompt	DAVIS cumulative prompt
T0	A cat in the grass in the sun.	A cyclist packs a bicycle inside a room.
T1	A dog in the grass in the sun. [P]	. . . The cyclist now wears a cobalt-blue jersey. [P]
T2	. . . The changed main subject has one thin white detail. [P]	. . . The jersey also has one thin white detail. [P]
T3	. . . one thin white detail and a subtly glossy finish. [P]	. . . one thin white detail and a subtly glossy finish. [P]
T4	. . . and a soft warm highlight. [P]	. . . and a soft warm highlight. [P]
T5	. . . with a few faint dust motes nearby. [P]	. . . with a few faint dust motes nearby. [P]

A.11 ADDITIONAL QUALITATIVE CASE STUDIES

Figures 4 and 5 provide two additional continuous-editing sessions. Each figure shows the initial state (T0) and five cumulative edits (T1–T5), comparing Exact (Wan-VACE), HetCache, EasyCache and ALIGNFLOW. Per-turn fidelity metrics, generation times and speedups are included in the panels.

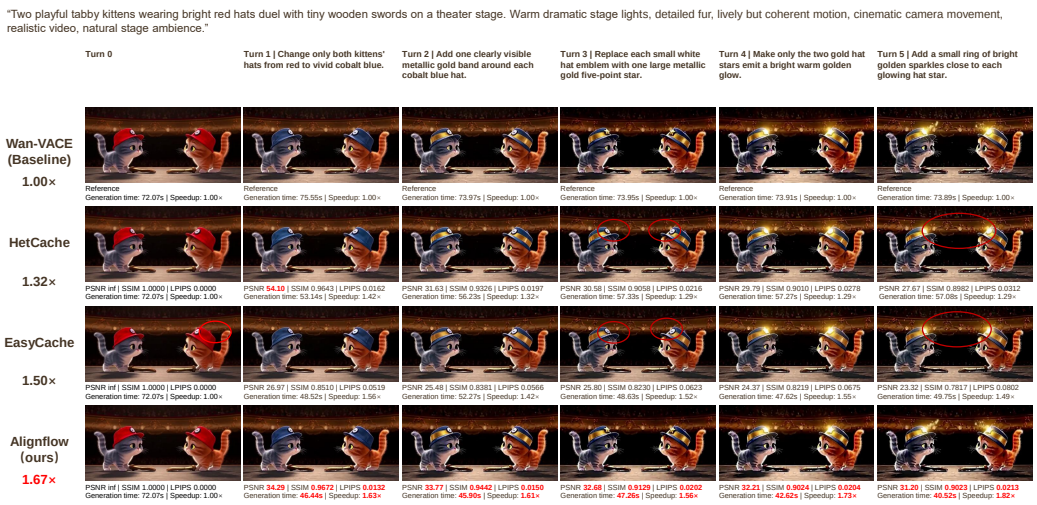


Figure 4: **Additional qualitative case study: kittens on a theater stage.** Five cumulative edits change the hats' color, add gold bands and stars, and introduce glow and sparkles.

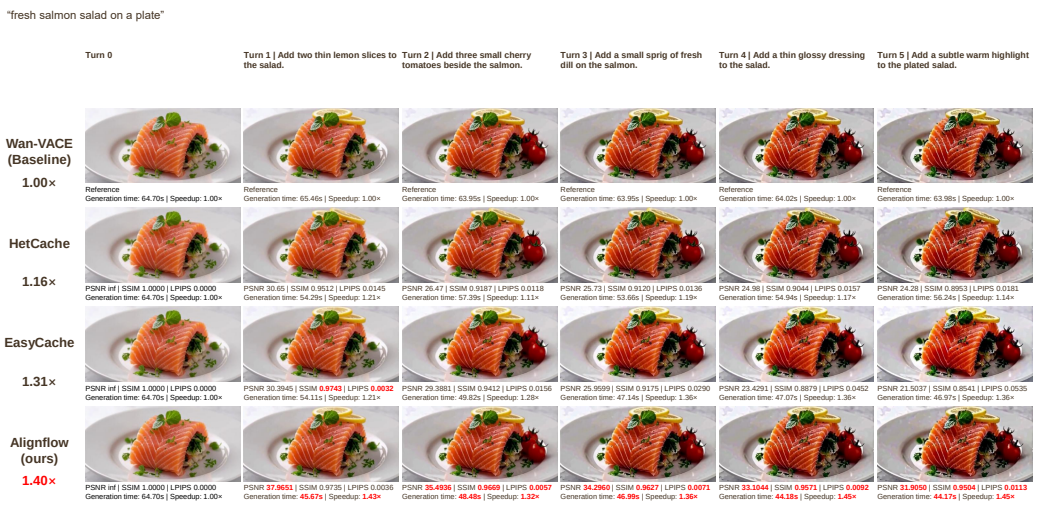


Figure 5: **Additional qualitative case study: salmon salad.** Five cumulative edits add lemon slices, cherry tomatoes, dill, glossy dressing and a warm highlight.